

5B – Autostrada

autor zadania: Kamil Dębowski

Treść

Dany jest opis trzypasmowej autostrady. Dla każdego pasa znamy pozycje oraz prędkość samochodów, które się na nim znajdują. Oznaczamy te prędkości przez v_1, v_2 oraz v_3 , dla odpowiednio pierwszego, drugiego oraz trzeciego pasa. Znamy również maksymalną prędkość v_0 naszego samochodu. Jego prędkość możemy dowolnie zmieniać, jednak musi ona pozostać w przedziale $[0, v_0]$. Możemy również w pomijalnym czasie zmieniać pas ruchu na sąsiedni. Wszystkie samochody, włącznie z naszym, mają tę samą długość. Odległości między samochodami w jednym pasie są wielokrotnościami długości samochodu. Oczywiście nie możemy zderzyć się z żadnym innym samochodem. Jesteśmy pytani o pierwszy moment, w którym możemy wyprzedzić wszystkie samochody.

Rozwiązanie

Do rozwiązania zadania użyjemy programowania dynamicznego. Dla każdego pasa będziemy określać pozycję względem samochodów, które na nim jadą, a nie względem początku trasy. Będziemy stopniowo zwiększać czas t , zaczynając z $t = 0$. Dla każdego pasa będziemy pamiętać najbardziej wysuniętą na prawo pozycję na tym pasie, na której możemy się znaleźć w czasie t . Oznaczmy te pozycje przez p_1, p_2 oraz p_3 . Zwróćmy uwagę, że wartości te nie muszą być całkowitoliczbowe.

Każda z tych wartości może zmieniać się z pochodną $v_0 - v_i$ albo z pochodną 0 względem czasu. Oznacza to, że na każdym pasie albo jedziemy ze swoją maksymalną prędkością (i przez czas t' zmieniamy swoją pozycję o $t' \cdot (v_0 - v_i)$) albo jedziemy równo z innymi samochodami, ponieważ nie możemy wjechać w samochód jadący bezpośrednio przed nami. Musimy zatem umieć wykrywać kiedy będziemy musieli zwolnić ze względu na następny samochód przed nami.

Inną rzeczą, którą musimy umieć wykrywać, jest możliwość zmiany pasa. Jeśli dla uporządkowanej pary sąsiednich pasów (i, j) na pasie i możemy być dalej niż na pasie j (względem początku trasy) i możemy zjechać z pasa i na pas j , to musimy rozważyć tę opcję. W celu aktualizacji czasu t oraz wartości p_i wybieramy oczywiście wydarzenie, które nastąpi jako pierwsze.

W celu wykrywania momentów, które musimy wziąć pod uwagę, możemy użyć odpowiednich struktur danych, odpowiednio dostosowanego wyszukiwania binarnego, lub jeśli jesteśmy wystarczająco ostrożni, znaleźć je w amortyzowanym czasie liniowym.

Gdy na jakimś pasie znajdziemy się przed wszystkimi innymi samochodami, opłacalnym jest do końca jechać ze swoją maksymalną prędkością aż wyprzedzimy wszystkie inne samochody. Wtedy znamy dokładny wynik.

Alternatywne podejście płynie z zauważenia, że w miarę opłaca nam się trzymać środkowego pasa i jechać na nim do przodu. Jeśli jesteśmy na środkowym pasie i możemy przejechać o pole do przodu, to robimy to z naszą maksymalną prędkością. W przeciwnym wypadku chcemy dostać się na następne wolne pole na środkowym pasie. Mamy tutaj dwie możliwości – użyć do tego pierwszego lub trzeciego pasa. Dla każdego z nich możemy wykryć najbliższy moment, w którym jest to możliwe. W przypadku pierwszego – szybszego pasa, interesuje nas jedynie czy możemy od razu na niego zjechać. Jeśli tak, natychmiastowo to robimy i jedziemy nim (albo z naszą maksymalną prędkością albo tuż za samochodem z jego prędkością) tak długo, aż będziemy mogli zjechać na środkowy pas. W przypadku trzeciego – wolniejszego pasa, interesuje nas najbliższy przedział przed nami, którego długość pozwoli nam dotrzeć do następnego wolnego pola na środkowym pasie – zakładając, że przejedziemy ten przedział z naszą maksymalną prędkością.

W przypadku pierwszego pasa wygodnie jest użyć wyszukiwania binarnego, a w przypadku trzeciego pasa – drzewa przedziałowego. Również w tym rozwiązaniu dostateczna ostrożność pozwala obyć się bez struktur danych i rozwiązać zadanie w amortyzowanym czasie liniowym.

By nie musieć walczyć z dokładnością obliczeń zmiennoprzecinkowych, warto zauważyć, że wszystkie interesujące wydarzenia mają miejsca w momentach czasu, które po zapisaniu jako liczba wymierna mają mianownik będący dzielnikiem $(v_0 - v_1)(v_0 - v_2)(v_0 - v_3)(v_1 - v_2)(v_1 - v_3)(v_2 - v_3)$. Oznaczmy wspomniany iloczyn przez M . Warto zatem nie trzymać w ogóle liczb zmiennoprzecinkowych, a pomnożyć liczony czas przez M . Alternatywnie, można trzymać momenty w czasie jako ułamki o mianowniku równym M . Ze względu na małe limity na wartości v_i nie powinno to sprawiać problemów ze zmieszczeniem się w 64-bitowych typach zmiennych.

Jeśli poprawnie zaimplementowane, otrzymujemy rozwiązanie działające w złożoności $\mathcal{O}(L \cdot \log(L))$ lub nawet $\mathcal{O}(L)$.